# A Measure of Space for Computing over the Reals

Paulin Jacobé de Naurois

# A Measure of Space for Computing over the Reals

Paulin Jacobé de Naurois*

LIPN - Université Paris XIII
99, avenue Jean-Baptiste Clément
93430 Villetaneuse - FRANCE
email: denaurois@lipn.univ-paris13.fr

**Abstract.** We propose a new complexity measure of space for the BSS model of computation. We define $\mathsf{LOGSPACE}_W$ and $\mathsf{PSPACE}_W$ complexity classes over the reals. We prove that $\mathsf{LOGSPACE}_W$ is included in $\mathsf{NC}^2_{\mathbb{R}} \cap \mathsf{P}_W$, i.e. is small enough for being relevant. We prove that the Real Circuit Decision Problem is $\mathsf{P}_{\mathbb{R}}$-complete under $\mathsf{LOGSPACE}_W$ reductions, i.e. that $\mathsf{LOGSPACE}_W$ is large enough for containing natural algorithms. We also prove that $\mathsf{PSPACE}_W$ is included in $\mathsf{PAR}_{\mathbb{R}}$.
**Keywords:** BSS model of computation, weak model, algebraic complexity, space.

## Introduction

The real number model of computation, introduced in 1989 by Blum, Shub and Smale in their seminal paper [BSS89], has proved very successful in providing a sound framework for studying the complexity of decision problems dealing with real numbers. A large number of complexity classes have been introduced, and many natural problems have been proved to be complete for these classes. A nice feature of this model is that it extends many concepts of the classical complexity theory to the broader setting of real computation; in particular a question $\mathsf{P}_{\mathbb{R}} \neq \mathsf{NP}_{\mathbb{R}}$ has arisen, which seems at least as difficult to prove as the classical one, and several $\mathsf{NP}_{\mathbb{R}}$-complete natural problems have been exhibited.

It has been soon pretty obvious, however, that all features of the classical complexity theory could not be brought to this setting. In particular, the only complexity measures considered so far were dealing with *time*, and not, say, *space*: in 1989, Michaux proved in [Mic89] that, under a straightforward notion of space, everything is computable in constant space. Therefore, no notion of logarithmic or polynomial space complexity exists so far over the reals. A way to deal with this situation has been to define parallel complexity classes in terms of algebraic circuits, such that the $\mathsf{NC}^i_{\mathbb{R}}$ and the $\mathsf{PAR}_{\mathbb{R}}$ classes.

This model of computation has also long been criticized for being unrealistic: the assumption that one could multiply two arbitrary real numbers in constant

* Partially supported by the ANR project NO CoST: New tools for complexity - semantics and types

time was the usual target, that Koiran faced in [Koi97] by defining a notion of *weak* cost that increases the cost for repeatedly multiplying or adding numbers.

Inspired by his approach, we propose here a new measure of space for the real number model, denoted as *weak* space, such that a repeated sequence of multiplications or additions on a number increases its size. Our notion allows us to define a logarithmic space complexity class, that falls within $\mathsf{NC}^2_{\mathbb{R}}$. We also prove that this class is large enough for containing natural algorithms: in particular, we prove a $\mathsf{P}_{\mathbb{R}}$-completeness result under $\mathsf{LOGSPACE}_W$ reductions.

The paper is organized as follows: in Section 1, we recall concepts and notations from the BSS model of computation. We define machines, circuits, and some major complexity classes. In Section 2, we briefly recall Michaux's result, and sketch a proof. In Section 3, we briefly introduce Koiran's notion of weak cost, and state some of the major results related to this notion. Then, we introduce our notion of *weak size* in Section 4, and state our results.

## 1 A Short Introduction on the BSS Model

In this section, we list the notations used in the paper, and recall some basic notions and results on the BSS model. A comprehensive reference for these notions is [BCSS98].

### 1.1 Notations

For an integer $c \in \mathbb{Z}$ we define its height as $\lceil \log(|c|+1) \rceil$. The height of an integer is the number of digits of its binary encoding. We also define $\mathbb{R}^* = \bigcup_{n\in\mathbb{N}} \mathbb{R}^n$.

### 1.2 Real Machines

We consider BSS machines over $\mathbb{R}$ as they are defined in [BSS89,BCSS98]. Roughly speaking, such a machine takes an input from $\mathbb{R}^*$ , performs a number of arithmetic operations and comparisons following a finite list of instructions, and halts returning an element in $\mathbb{R}^*$ (or loops forever). Such a machine can be seen as a Turing machine over $\mathbb{R}$. It essentially consists in a finite directed graph, whose nodes are *instructions*, together with an input tape, an output tape, and a bi-infinite work tape, equipped with scanning heads. The instructions can be of the following types: *Start*, *Input* (reads an input value), *Output* (writes an output value), *Computation* (performs one arithmetical operation on two elements on the work tape), *Constant* (writes a constant parameter $A_i \in \mathbb{R}$), *Branch* (compares two elements, and branches accordingly), *Shift*, *Copy* and *Halt*.

For a given machine $M$, the function $\varphi_M$ associating its output to a given input $x \in \mathbb{R}^*$ is called the *input-output function*. We say that a function $f : \mathbb{R}^* \to \mathbb{R}^*$ is *computable* when there is a machine $M$ such that $f = \varphi_M$.

Also, a set $L \subseteq \mathbb{R}^*$, or a *language* is *decided* by a machine $M$ if its characteristic function $\chi_L : \mathbb{R}^* \to \{0, 1\}$ coincides with $\varphi_M$.

This model of computation allows one to define complexity classes. In particular, $\mathsf{P}_{\mathbb{R}}$ is the set of subsets of $\mathbb{R}^*$ that are decided by a real machine that works in deterministic polynomial time. Similarly, $\mathsf{NP}_{\mathbb{R}}$ is the set of subsets of $\mathbb{R}^*$ that are decided by a real machine that works in nondeterministic polynomial time i.e, $x \in \mathbb{R}^*$ is accepted if and only if there exists $y \in \mathbb{R}^*$ of polynomial size such that the machine accepts $(x, y)$.

### 1.3 Configurations

**Definition 1.** Configurations

- *A* configuration *of a machine $M$ is given by an instruction $q$ of $M$ along with the position of the heads of the machine and three words $w_{input} \in \mathbb{R}^*$, $w_{work} \in \mathbb{R}_*$, $w_{output} \in \mathbb{R}^*$ that give the contents of the input tape, of the work tape and of the output tape.*
- *A* transition *of a machine $M$ is a couple $(c_i, c_j)$ of configurations such that, whenever $M$ is in configuration $c_i$, $M$ reaches $c_j$ in one computation step.*

**Definition 2.** Configuration Graph
*For a given machine $M$ and a given set $\mathcal{C}$ of configurations of $M$, we define the* configuration graph *of $M$ on $\mathcal{C}$ to be the directed graph with vertexes all elements in $\mathcal{C}$, and edges all transitions of $M$ between elements in $\mathcal{C}$.*

### 1.4 Algebraic Circuits

We introduce the notion of algebraic circuits, that allows to denote parallel computations and to define complexity classes below $\mathsf{P}_{\mathbb{R}}$.

**Definition 3.** Algebraic Circuit
*An* algebraic circuit *$\mathcal{C}$ is a sequence of gates $(G_1, \ldots, G_m)$ of one of the following types:*

1. *Input gates: $G_i = x_i$, takes the input $x_i$ from $\mathbb{R}$,*
2. *Arithmetic gates: perform the operation $*$ to the outputs of gates $G_j$ and $G_l$, $j, l < i$ and $* \in \{+, -, ., /\}$,*
3. *Constant gates: $G_i = A_i$, $A_i \in \mathbb{R}$,*
4. *Sign gates: If $G_j \geq 0$ then $G_i = 1$ else $G_i = 0$, $j < i$.*

*If a circuit has $n$ input gates, we can suppose that they are the first ones, $G_1, \ldots, G_n$. If moreover the last node $G_m$ is a sign node, we shall say that $\mathcal{C}$ is a* decision circuit*.*

An algebraic circuit is a finite directed graph with no loops: its *size* is the number of gates, and its *depth* is the length of its longest path, starting from an input gate.

Algebraic circuits extend the classical notion of boolean circuit to the BSS setting, and allows one to define the $\mathsf{NC}_{\mathbb{R}}$ hierarchy of complexity classes:

**Definition 4.** $\mathsf{NC}_{\mathbb{R}}$
*For all $i \in \mathbb{N}$, $\mathsf{NC}^i_{\mathbb{R}}$ is the class of real decision problems decided by a $\mathsf{P}$-uniform family of circuits of polynomial size and of depth bounded by $\mathcal{O}(\log^i(n))$, and*

$$\mathsf{NC}_{\mathbb{R}} = \bigcup_{i \in \mathbb{N}} \mathsf{NC}^i_{\mathbb{R}}.$$

As remarked by Poizat in [Poi95], in the definition above the uniformity of the family can be considered relative to the classical Turing model. Hence, a $\mathsf{P}$-uniform family of algebraic decision circuit is such that there exists an finite enumeration of all the constant gates in the family. There exists then a $\mathsf{P}$ time Turing machine which, on input $n, k$, outputs a discrete description of the $k^{\text{th}}$ gate of the $n^{\text{th}}$ circuit of the family.

**Proposition 1.** *[Cuc92]*

$$\mathsf{NC}_{\mathbb{R}} \subsetneq \mathsf{P}_{\mathbb{R}}.$$

## 2 Michaux's Result

This section is devoted to a brief exposition of Michaux's Result [Mic89], which states that a straightforward measure of space fails in differentiating one algorithm from another. In this section, we will use the following notion of space as a complexity measure:

**Definition 5.** Unit Space
*Let $M$ be a machine over $\mathbb{R}$, and let $c$ be a configuration of $M$. We define $USize(c)$, the* unit size *of $c$ to be number of non-empty cells on the work tape at configuration $c$. Assume that on an input $(x_1, \ldots, x_n)$, the computation of $M$ ends within $t$ computation steps. The computation follows a path $c_0, \ldots, c_t$. We define the* unit space *used by $M$ on input $(x_1, \ldots, x_n)$ to be*

$$USpace(M, (x_1, \ldots, x_n)) = \max_{0 \leq k \leq t} USize(c_k).$$

*Assume that the running time of $M$ is bounded by a function $t$. We define the* unit space *used by $M$ on input size $n$ to be*

$$USpace(M, n) = \max_{(x_1, \ldots, x_n) \in \mathbb{R}^n} USpace(M, (x_1, \ldots, x_n)).$$

This notion of *unit space* is essentially the same as the classical notion of space for Turing machines. While in the classical Turing model this notion gives rise to a whole hierarchy of complexity classes like $\mathsf{LOGSPACE}$, $\mathsf{PSPACE}$, interlaced with the time hierarchy, this is not the case in the real setting. In order to precise a bit how unit space behaves on the reals, let us begin with the following well known technical result.

**Lemma 1.** *Let $M$ be a real machine with parameters $A_1, \ldots, A_m$, whose running time is bounded by a function $t$. Let $n \in \mathbb{N}$. On any input $x_1, \ldots, x_n \in \mathbb{R}^n$, at any computation step $k \leq t(n)$, any non-empty cell on the work tape, say $e_l$, contains the evaluation of a rational fraction $f_{l,k} \in \mathbb{Z}(X_1, \ldots, X_{n+m})$ on $(x_1, \ldots, x_n, A_1, \ldots, A_m)$.*

*Proof.* Details arguments can be found in [Mic89,Poi95,Koi97]. We only sketch a proof here. The key argument is that, at any computation step $k$, the content of any cell $e_l$ is obtained from the input values and the parameter values by a finite sequence of arithmetical operations. Therefore, the value in $e_l$ is the evaluation of a rational fraction $f_{l,k} \in \mathbb{Z}(X_1, \ldots, X_{n+m})$ on $(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

**Proposition 2.** *[Mic89] Let $L \subseteq \mathbb{R}^*$ by a real language decided by a machine $M$ in time bounded by a function $t$. There exists a constant $k \in \mathbb{N}$ and a machine $M'$ deciding $L$ in unit space $k$.*

*Proof.* We only sketch the proof here. The interested reader can find more explanations in [Mic89,Poi95].

Rational fractions with integer coefficients can be easily encoded in binary, therefore, by Lemma 1, any configuration of $M$ can also be encoded in binary. It suffices to realize that this binary encoding can be embedded into the digits of only two real numbers. Then, there exists a machine $M'$ simulating $M$ with only a constant number of real registers, among which two are needed for encoding the configurations of $M$.

## 3 The Weak BSS Model by Koiran

### 3.1 Definitions

**Definition 6.** Weak Cost
*Let $M$ be a machine whose running time is bounded by a function $t$, and let $A_1, \ldots, A_m$ be its real parameters. On any input $x_1, \ldots, x_n$, the computation of $M$ consists in a sequence $c_0, \ldots, c_t, t \leq t(n)$ of configurations. To a transition $c_k, c_{k+1}$ in this sequence we associate its* weak cost *as follows:*

- *If the current instruction of $c_k$ is a computation node, let $e_l$ be the current cell on the work tape: the transition $c_k, c_{k+1}$ consists in the computation of a rational fraction $f_{l+1,k+1} = g_{l+1,k+1}/h_{l+1,k+1} \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$, which is placed on the cell $e_{l+1}$ in $c_{k+1}$. The* weak cost *of the transition $c_k, c_{k+1}$ is defined to be the maximum of deg($g_{l+1,k+1}$), deg($h_{l+1,k+1}$), and the maximum height of the coefficients of $g_{l+1,k+1}$ and $h_{l+1,k+1}$.*
- *Otherwise, the* weak cost *of the transition $c_k, c_{k+1}$ is defined to be 1.*

*The* weak running time *of $M$ on input $x_1, \ldots, x_n$ is the sum of the weak costs of the transitions in the sequence $c_0, \ldots, c_t$.*

*The* weak running time of $M$ *is the function that associates with every $n$ the maximum over all $x \in \mathbb{R}^n$ of the running time of $M$ on $x$.*

### 3.2 Some Results

**Lemma 2.** *[Koi97] A function is polynomial-time in the weak BSS model if and only if it is polynomial-time computable in the standard BSS model and the rational fractions $f_{l,k}$ have polynomial degree and coefficients of polynomial bit-size.*

Let $\mathsf{P}_W$ (respectively $\mathsf{NP}_W$) be the set of real languages decided in deterministic (resp. nondeterministic) weak polynomial time, and $\mathsf{EXP}_W$ be the set of real languages decided in weak exponential time by a real machine.

**Proposition 3.**

$$\mathsf{NC}^2_{\mathbb{R}} \not\subset \mathsf{P}_W \subsetneq \mathsf{P}_{\mathbb{R}} \subseteq \mathsf{NP}_W = \mathsf{NP}_{\mathbb{R}} \subseteq \mathsf{PAR}_{\mathbb{R}} \subseteq \mathsf{EXP}_W.$$

$\mathsf{P}_W \subsetneq \mathsf{NP}_W = \mathsf{NP}_{\mathbb{R}}$ is from [CSS94], $\mathsf{NP}_{\mathbb{R}} \subseteq \mathsf{EXP}_W$ from [Koi97] (where it is shown that the inclusion is strict). The missing items can be found in [BCSS98].

## 4 Weak Size and Space

### 4.1 Definitions

Instead of considering a unit size for all values on the work tape, which allows one to decide every decidable language in constant space, we would like to have a notion of size for the values computed on the work tape. The weak size of a computed value is a reasonable upper bound for the size of a boolean description of the corresponding rational fraction with integer coefficients. The weak size of a configuration is then the sum of the weak sizes of all computed values on the work tape in this configuration.

Yet, we need to precise a bit more the idea. Our purpose is to have a "nice" measure of space, allowing one to define a reasonable logarithmic space class. A trivial rational fraction like $f_1(X_1, \ldots, X_n, A_1, \ldots, A_m) = X_1$ has clearly a boolean description of size 1, while, for describing $f_n(X_1, \ldots, X_n, A_1, \ldots, A_m) = X_n$, one would need $\lceil \log(n+1) \rceil$ digits (for encoding the variable index). It seems rather unsatisfactory that a logarithmic space configuration may have a logarithmic number of occurrences of $f_1$, but only a constant ones of $f_n$. This feature can be corrected by allowing a permutation of the input variables, provided the permutation is simple enough, i.e can be described in logarithmic boolean space. In this paper, we have restricted ourselves to circular permutations, that can be described by an offset in $\{0, \ldots, n-1\}$.

**Definition 7.** Weak Size
*Assume $A_1, \ldots, A_m \in \mathbb{R}^m$ are given real numbers, and let $g \in \mathbb{Z}[X_1, \ldots, X_{n+m}]$ be a real polynomial with integer coefficients. Define a real polynomial $g_{A_1,\ldots,A_m} = g[X_1, \ldots, X_n, A_1, \ldots, A_m]$, with free variables $X_1, \ldots, X_n$. Let $0 \leq \mathsf{O} < n$, $\mathsf{O} \in \mathbb{N}$ be a number, the* offset. *To $g$, $A_1, \ldots, A_m \in \mathbb{R}^m$ and $\mathsf{O}$, we associate the following:*

– *$deg(g)$ is the degree of $g$. We will write $\mathsf{D}(g)$ for $\lceil \log(deg(g)+1) \rceil$.*
– *$Var_{A_1,\ldots,A_m}(g) \subseteq \{X_1,\ldots,X_n\}$ is the set of input variables on which $g_{A_1,\ldots,A_m}$ effectively depends.*
– *$R_{A_1,\ldots,A_m,\mathsf{O}}(g) = \max\{i+\mathsf{O} \mod n\}$ for $X_i \in Var_{A_1,\ldots,A_m}(g)$, is the* range *of $g$. We will write $\mathsf{R}(g)$ for $\lceil \log(R_{A_1,\ldots,A_m,\mathsf{O}}(g)+1) \rceil$.*
– *$\mathsf{N}(g) \in \mathbb{N}$ is the number of non-zero monomials of $g$.*
– *$S(g) \in \mathbb{Z}$ is the maximal absolute value of the integer coefficients of $g$. We will write $\mathsf{S}(g)$ for $\lceil \log(2S(g)+1) \rceil$.*
– *$Vc_{A_1,\ldots,A_m}(g)$ is the maximum, for every monomial of $g$, of the number of input variables on which it effectively depends. We will write $\mathsf{V}(g)$ for $Vc_{A_1,\ldots,A_m}(g)$.*

*The weak size $S_{A_1,\ldots,A_m,\mathsf{O}}(g)$ of $g$ is defined as follows:*

$$S_{A_1,\ldots,A_m,\mathsf{O}}(g) = \mathsf{N}(g)\,(\mathsf{S}(g) + \mathsf{V}(g).\mathsf{R}(g) + \mathsf{V}(g).\mathsf{D}(g)) \tag{1}$$

*For a rational fraction $f = g/h$ we take the weak size of $f$ to be the maximum of the weak sizes of $g$ and $h$.*

It is clear that the weak size of $g$ bounds the size of a boolean encoding of $g$, where $g$ is presented as a sum of monomials modulo a circular permutation of the variable indexes. We do not take into account succinct boolean descriptions of factorized polynomials to ensure the tractability of our measure.

This measure of size for an element on the work tape naturally yields a notion of weak space for the given work tape, as follows:

**Definition 8.** Weak Space
*Let $M$ be a machine with real parameters $A_1,\ldots,A_m$. Let $c_k$ be a configuration of $M$, with the corresponding input $x_1,\ldots,x_n \in \mathbb{R}^n$. We define:*

– *$e_i,\ldots,e_j$ to be the non-empty part of the work tape in the configuration $c_k$.*
– *For any non-empty cell $e_l$ in $c_k$, we denote by $f_{l,k} \in \mathbb{Z}(x_1,\ldots,x_n,A_1,\ldots,A_m)$ the rational fraction it contains.*

*The weak size of the work tape at the configuration $c_k$ is then:*

$$Size_w(c_k) = \min_{0 \le \mathsf{O} < n} \sum_{l=i}^{j} S_{A_1,\ldots,A_m,\mathsf{O}}(f_{l,k})$$

*Assume that the running time of $M$ is bounded by a function $t$. For a given input $x_1,\ldots,x_n$, the computation of $M$ consists in a sequence $c_0,\ldots,c_t, t \le t(n)$ of configurations.*

*The* weak running space *of $M$ on input $x_1,\ldots,x_n$ is the maximum for all configurations $c_0,\ldots,c_t$ of their weak size.*

*The* weak running space *of $M$ is the function that associates with every $n$ the maximum over all $x \in \mathbb{R}^n$ of the running space of $M$ on $x$.*

**Definition 9.** Complexity Classes

- *A language $L \subseteq \mathbb{R}^*$ is in $\mathsf{LOGSPACE}_W$ if and only if there exist a machine $M$ and a constant $k \in \mathbb{N}$ such that, for all $n \in \mathbb{N}$, on input $x \in \mathbb{R}^n$, $M$ decides whether $x \in L$ in weak space less than $k\log(n)$.*
- *A language $L \subseteq \mathbb{R}^*$ is in $\mathsf{PSPACE}_W$ if and only if there exist a machine $M$ and two constants $k, d \in \mathbb{N}$ such that, for all $n \in \mathbb{N}$, on input $x \in \mathbb{R}^n$, $M$ decides whether $x \in L$ in weak space less than $kn^d$.*
- *A function $f : \mathbb{R}^* \to \mathbb{R}^*$ is in $\mathsf{FLOGSPACE}_W$ if and only if there exist a machine $M$ and two constant $k, m \in \mathbb{N}$ such that, for all $n \in \mathbb{N}$, on input $x \in \mathbb{R}^n$, and computation $c_0, \ldots, c_t$ of $M$ on $x$:*
  1. *$M$ computes $f(x)$ in weak space less than $k\log(n)$.*
  2. *for every configuration $c_i$ with current node an output node and current cell $e_l$, the weak size of the content of $e_l$ is less than $m$.*

In the definition of $\mathsf{FLOGSPACE}_W$, the output consists in a sequence of real values of constant weak size in the input. This ensures that one can compose $\mathsf{FLOGSPACE}_W$ algorithms, and that the result of the composition remains an $\mathsf{FLOGSPACE}_W$ algorithm. This is necessary for defining notions like logarithmic space reductions and for obtaining completeness results.

### 4.2 What Michaux's Result Becomes

**Lemma 3.** *There exists $L \subseteq \mathbb{R}^*$ such that:*

- *$L \in \mathsf{P}_W$*
- *for all $k \in \mathbb{N}$, $L$ is not decidable in weak space less than $k$.*

*Proof.* Let $p(X_1, \ldots, X_n) = X_1 + \ldots + X_n$, and consider the set $L$ of points $(x_1, \ldots, x_n) \in \mathbb{R}^n$, such that $p(x_1, \ldots, x_n)$ equals 0. Assume $L$ is decided by a machine $M$. It is well known that the set of inputs accepted by a BSS machine is semi-algebraic, therefore, $L$ can be described as a finite union of sets given by systems of polynomials inequalities of the form

$$\bigwedge_{i=0}^{s} F_i(X_1, \ldots, X_n) = 0 \wedge \bigwedge_{j=0}^{t} G_j(X_1, \ldots, X_n) > 0,$$

where the values $F_i(x_1, \ldots, x_n)$ and $G_j(x_1, \ldots, x_n)$ are effectively computed by $M$. Since $L$ has dimension $n-1$, at least one of these sets must have dimension $n-1$. Since the set described by the $G_j's$ is open, it must be nonempty, and then it defines an open subset of $\mathbb{R}^n$. All the polynomials $F_i$ vanish on that nonempty open subset of $L$. Since this open subset of $L$ is clearly infinite, and $p$ is an irreducible polynomial, all the polynomials $F_i$ must vanish on the whole set $L$. It is then a well known result ([BCSS98], Proposition 2 p.362) that the polynomials $F_i$ are multiples of $p$. Also, at least one of these $F_i$ is a non-trivial multiple of $p$.

It is clear that $p(x_1, \ldots, x_n)$ has weak size at least $n\log(n)$, and so does this non-trivial multiple of $p$. Therefore, $M$ decides $L$ in weak space at least $n\log(n)$.

### 4.3 Structural Complexity Results

**Theorem 1.**

$$\mathsf{LOGSPACE}_W \subseteq \mathsf{P}_W \cap \mathsf{NC}^2_{\mathbb{R}},$$
$$\mathsf{PSPACE}_W \subseteq \mathsf{PAR}_{\mathbb{R}}.$$

*Proof.* In a first step, we prove $\mathsf{LOGSPACE}_W \subseteq \mathsf{P}_{\mathbb{R}}$. The key argument is an upper bound for the number of configurations of weak size at most $k\log(n)$. Consider a machine $M$, with $t$ nodes. For a fixed input size $n$, and an offset $\mathsf{O}$, simple counting arguments show that the number of rational fractions of weak size at most $B$ for some $B \in \mathbb{N}$ is bounded by $\alpha^B$, for some $\alpha \in \mathbb{N}$. It follows that the number of possible work tape contents of weak size $B$, for the same fixed offset, is bounded by $(2\alpha^2)^B$. Taking into account all possible values for the offset, the scanning head positions and the current node of the machine, the number of configurations of weak size at most $B$ is then bounded by $tn^2B(2\alpha^2)^B$. When $B = k\log(n)$, this bound is polynomial.

$\mathsf{LOGSPACE}_W \subseteq \mathsf{P}_W$ follows then by Lemma 2, since all rational fractions of logarithmic weak size have clearly polynomial degrees and coefficient heights.

$\mathsf{LOGSPACE}_W \subseteq \mathsf{NC}^2_{\mathbb{R}}$ is then proven along the lines of [Bor77]: given a $\mathsf{LOGSPACE}_W$ machine $M$, we exhibit a $\mathsf{NC}^1_{\mathbb{R}}$ construction of its configuration graph. This construction involves some numeric computation, in order to check whether two given configurations are connected, and produces a boolean description of the configuration graph of $M$. Next, it suffices to decide whether the input and accepting configurations are connected in this graph: this is the classical reachability problem, which is decidable in the boolean class $\mathsf{NC}^2$.

$\mathsf{PSPACE}_W \subseteq \mathsf{PAR}_{\mathbb{R}}$ is a corollary.

### 4.4 Completeness Results

**Definition 10.** *[CT92]* Real Circuit Decision Problem ($\mathsf{CDP}_{\mathbb{R}}$)
*Input:* $(\mathcal{C}, \overline{x})$*, where* $C$ *is an arithmetic circuit with* $k$ *input gates and* $\overline{x} \in \mathbb{R}^k$*.*
*Question: Does* $\mathcal{C}$ *output 1 on input* $\overline{x}$*?*

It has been shown in [CT92] that $\mathsf{CDP}_{\mathbb{R}}$ is $\mathsf{P}_{\mathbb{R}}$-complete under $\mathsf{NC}^2_{\mathbb{R}}$-reductions.

**Theorem 2.** $\mathsf{CDP}_{\mathbb{R}}$ *is* $\mathsf{P}_{\mathbb{R}}$*-complete under* $\mathsf{FLOGSPACE}_W$*-reductions.*

*Proof.* The proof follows [CT92]. The reduction happens to be in $\mathsf{FLOGSPACE}_W$.

We have stated this completeness results under $\mathsf{FLOGSPACE}_W$ reductions. By Theorem 1, it is clear that $\mathsf{FLOGSPACE}_W$ reductions are in $\mathsf{P}_W \cap \mathsf{NC}^2_{\mathbb{R}}$. The problem considered has already been proven complete under first-order reductions [GM96], which also happen to be in $\mathsf{P}_W \cap \mathsf{NC}_{\mathbb{R}}$. Yet, it remains unclear how the two types of reductions compare.

## 5 Concluding Remarks and Open Questions

In the discrete model, space has proven to be a very relevant complexity measure. Many natural problems have been found in $\mathsf{LOGSPACE}$, and many others in $\mathsf{NC}^2$ whose membership in $\mathsf{LOGSPACE}$ is unclear. We believe that weak space may play the same role in the real setting. An argument in this direction is the following remark: consider a real algorithm that reads an input, normalizes it to $\{0,1\}$ with some step function, and applies a boolean $\mathsf{LOGSPACE}$ procedure. Real complexity analysis until now only allowed one to say that such a real algorithm belongs to $\mathsf{P}_W \cap \mathsf{NC}^2_{\mathbb{R}}$: the algorithmic flavor behind it was lost. However, it is now clear that such an algorithm belongs to $\mathsf{LOGSPACE}_W$. An important task now is to exhibit some natural problems in $\mathsf{LOGSPACE}_W$. Others in $\mathsf{NC}^2_{\mathbb{R}}$ or $\mathsf{P}_W$, not easily in $\mathsf{LOGSPACE}_W$, may also be of interest.

Structural results remain also to be found. In particular, it needs to be checked whether the following conjecture holds:

*Conjecture 1.*

$$\mathsf{NC}^1_{\mathbb{R}} \not\subset \mathsf{LOGSPACE}_W,$$
$$\mathsf{LOGSPACE}_W \subseteq \mathsf{NC}^1_{\mathbb{R}} \Rightarrow \mathsf{LOGSPACE} \subseteq \mathsf{NC}^1.$$

Similar questions arise also for $\mathsf{PSPACE}_W$.


## Acknowledgements

We thank the anonymous referees for their helpful comments, and for pointing out references to the notion of first-order reductions.


## References


BCSS98. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

Bor77. A. Borodin. On relating time and space to size and depth. *SIAM J. Comp.*, 6:733–744, 1977.

BSS89. L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the Amer. Math. Soc.*, 21:1–46, 1989.

CSS94. F. Cucker, M. Shub, and S. Smale. Separation of complexity classes in Koiran's weak model. *Theoretical Computer Science*, 133(1):3–14, 11 October 1994.

CT92. F. Cucker and A. Torrecillas. Two p-complete problems in the theory of the reals. *Journal of Complexity*, 8(4):454–466, 1992.

Cuc92. F. Cucker. $\mathrm{P}_{\mathbb{R}} \neq \mathrm{NC}_{\mathbb{R}}$. *Journal of Complexity*, 8:230–238, 1992.

GM96. E. Grädel and K. Meer. Descriptive complexity theory over the real numbers. *Lecture Notes in Applied Mathematics*, 32:381–404, 1996.

Koi97. P. Koiran. A weak version of the blum, shub & smale model. *Journal of Computer and System Sciences*, 54:177–189, 1997.

Mic89. C. Michaux. Une remarque à propos des machines sur $\mathbb{R}$ introduites par Blum, Shub et Smale. *C. R. Acad. Sci. Paris*, 309, Série I:435–437, 1989.

Poi95. B. Poizat. *Les Petits Cailloux*. Aléas, 1995.